

А. В. Березняцкий

## РАЗРАБОТКА WEB-СЕРВЕРА ДЛЯ ХРАНЕНИЯ И УЧЕТА ЗАДАЧ ПОЛЬЗОВАТЕЛЯ

Статья посвящена разработке Web-сервера для хранения и учета задач пользователя на языке Java. Разработан сервер, генерирующий JSP-страницу в код HTML-страницы, которая видна пользователю. С этой страницы сервер получает запрос (регистрация, аутентификация, добавление задания и другое), обрабатывает и отправляет пользователю новую сгенерированную HTML-страницу с выполненным запросом. Для хранения логина, пароля и задач пользователя к серверу подключены базы данных MySQL.

Рассмотрим инструменты, технологии и другие решения. В силу того, что проект ограничен небольшим сроком реализации, было принято решение рассматривать и анализировать наиболее популярные средства в своей категории.

Для реализации серверной оставляющей был выбран язык программирования Java из-за своей надежности, простоты и кроссплатформенности он зарекомендовал себя как отличный выбор для Backend разработки.

Объектно-ориентированный язык Java [1, с. 14], разработанный в компании Sun Microsystems в 1995 году для оживления графики на стороне клиента с помощью апплетов, в настоящее время используется для создания переносимых на различные платформы и операционные системы программ. Язык Java нашел широкое применение в Интернет-приложениях, добавив на статические и клиентские веб-страницы динамическую графику, улучшив интерфейсы и реализовав вычислительные возможности. Но объектно-ориентированная парадигма и кроссплатформенность привели к тому, что уже буквально через несколько лет после создания язык практически покинул клиентские страницы и перебрался на серверы.

MySQL – это одна из самых популярных и самых распространенных СУБД (система управления базами данных) в интернете, ее применение идеально для интернет сайтов, как небольших, так и достаточно крупных. MySQL отличается хорошей скоростью работы, надежностью, гибкостью.

Java Server Pages представляет технологию, которая позволяет создавать динамические веб-страницы. По сути Java Server Page или JSP представляет собой html-код с вкраплениями кода Java. В то же время станицы jsp – это не стандартные html-страницы. Когда приходит запрос к определенной странице JSP, то сервер обрабатывает ее, генерирует из нее код html и отправляет его клиенту. В итоге пользователь после обращения к странице JSP видит в своем браузере обычную html-страницу. Как и обычные статические веб-страницы, файлы JSP необходимо размещать на веб-сервере. На рисунке 1 приведено схематично как работает JSP.

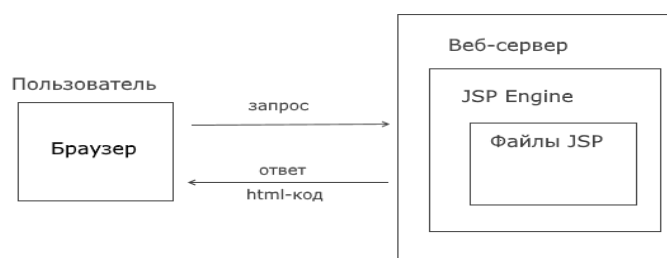


Рисунок 1 – Схема работы JSP

Для работы сервера нужно создать сервлеты [1, с. 456] для обработки и выполнения запросов пользователя, начальную html-страницу [2] и 3 JSP-страницы [1, с. 485]: для регистрации, аутентификации и отображения задач пользователя.

Для работы сервера к нему подключаем СУБД MySQL [1, с. 793]. В базе данных создадим 2 таблицы: для хранения логинов и паролей и для хранения задач пользователей. Связь таблиц осуществляется по прицепу: пользователь получает свой уникальный ID в первой таблице при регистрации и при записи задачи во вторую таблицу, она также получает такой ID. Когда сервер считывает задачи с базы данных, задачи, у которых ID не такой как у пользователя, пропускаются.

Для того, чтобы зайти в приложение, пользователь должен зарегистрироваться. Таким образом, начальная страница приложения предлагает пользователю зарегистрироваться либо аутентифицироваться. При нажатии на соответствующую ссылку (или кнопку) пользователю показывается либо форма для регистрации, либо форма для ввода имени пользователя и пароля. После успешной регистрации или аутентификации пользователь переходит на страницу со своим списком задач, страница представлена на рисунке 2.

Во избежание недоразумений, рисунок 1 сделан 9 мая 2019 года и сервер запускается локально, так как для запуска во всемирную паутину нужно иметь фиксированный IP-адрес и купить доменное имя.

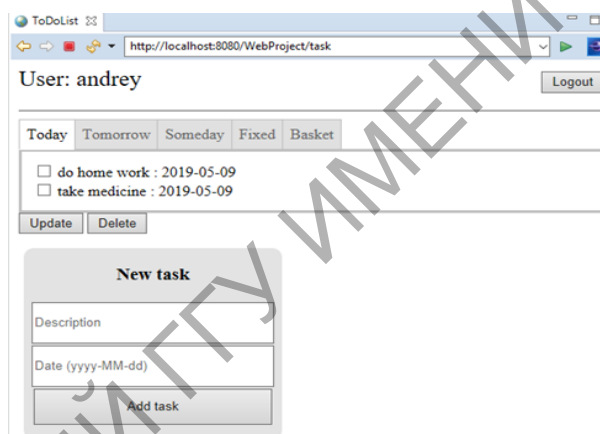


Рисунок 2 – Кабинет пользователя

Невыполненные задачи содержатся в разделах: Today, Tomorrow и Someday. Если задача добавляется в раздел Today или Tomorrow, то ей автоматически присваивается соответственно сегодняшняя или завтрашняя дата выполнения, все остальные задачи хранятся в разделе Someday.

В разделе Today отображаются как сегодняшние задачи, так и просроченные. Все задачи, добавленные в раздел Tomorrow, на следующий день после добавления отображаются в разделе Today. Аналогично некоторые задачи из раздела Someday образуют новое содержимое раздела Tomorrow.

Задачи можно отмечать как выполненные, после чего они исчезают из списка. При этом они не удаляются физически, а показываются в разделе Fixed.

Из любого раздела можно удалять задачи. В результате выполнения данной операции они попадают в корзину (Basket – еще один раздел). Задачу можно восстановить из корзины. Далее задача отображается в разделе, соответствующем дате выполнения. Корзину можно очистить полностью либо удалить каждую задачу по отдельности. В таком случае задача удаляется.

Таким образом, разработан новый Web-сервер для хранения и учета задач пользователя. Данное бесплатное приложение не имеет рекламы и очень прост в использовании.

## Литература

1 Блинов И. Н. Java. Методы программирования: уч.-мет. пособие / И. Н. Блинов, В. С. Романчик. – Минск: изд-во «Четыре четверти», 2013. – 896 с.

2 Справочник HTML CSS [Электронный ресурс] – Режим доступа: <http://htmlbook.ru/> – Дата доступа: 01.03.2019.

УДК 004.7

*М. С. Березовский*

### РЕАЛИЗАЦИЯ ИГРОВОГО ПРИЛОЖЕНИЯ «GALAXY SHOOTER» С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА C#

*Статья посвящена разработке игрового приложения «Galaxy Shooter», позволяющего игроку поучаствовать в космической баталии в качестве пилота космического корабля. Оно имеет все необходимое для генерирования противников для игрока и подсчета очков при их уничтожении. Реализована возможность генерирования различных усилений для игрока с соответствующим звуковым сопровождением. Приложение реализовано с использованием игрового движка Unity3D и объектно-ориентированного языка C#. Игровое приложение работает под операционной системой Windows, без установки специализированных дистрибутивов.*

Настоящая работа посвящена разработке широко распространенной тематике игровых приложений. В ходе работы создано игровое приложение «Galaxy Shooter», которое взаимодействует с пользователем, принимая от последнего различные инструкции.

Игровое приложение имеет все необходимое для генерирования противников для игрока и подсчета очков при их уничтожении. Реализована возможность генерирования усилений для игрока, которые могут быть собраны последним. Данные усиления реализованы таким образом, что для добавления новых усилений необходимо иметь ассет и осуществить пару кликов, что предоставляет большую возможность для расширения разработанного игрового приложения. Схожим образом могут быть добавлены и новые противники. Все необходимые настройки можно провести из редактора, не осуществляя никаких изменений в исходном коде игрового приложения.

Любые события в игровом приложении (сбор усилений, уничтожения противника, открытие огня по противнику и т. д.) сопровождаются звуковыми эффектами. Также для игрока, кораблей противника и усилений, а также взаимодействия между ними были созданы анимации. Таким образом, увеличивается погружение игрока в происходящее в игровом приложении. Задний фон игрового экрана, представляющий собой открытый космос с различного рода космическими объектами (звезды, кометы, планеты и т. д.), генерируется процедурно.

В работе используется специальный пакет разработанный командой Unity [3], позволяющий применять различного рода эффекты для придания лучшего визуального вида полученному игровому приложению.

При запуске игрового приложения пользователь попадает в открытый космос (рисунок 1). В центре экрана расположен корабль игрока. Управление им осуществляется путем нажатия клавиш W, A, S, D или стрелок. В правом верхнем углу экрана расположен счетчик очков, набранных игроком. В левом верхнем углу экрана расположено количество жизней, оставшееся у игрока (в начале игры количество жизней равно 3).